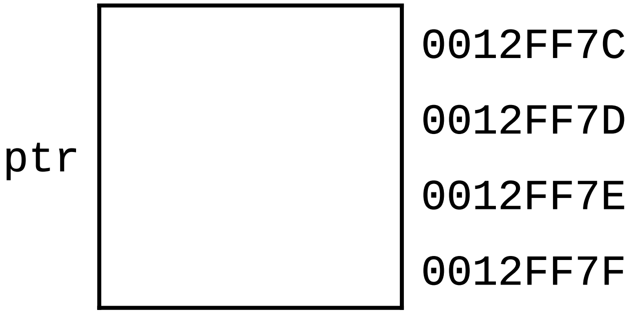
Dynamically allocate a variable



ptr

0012FF7C

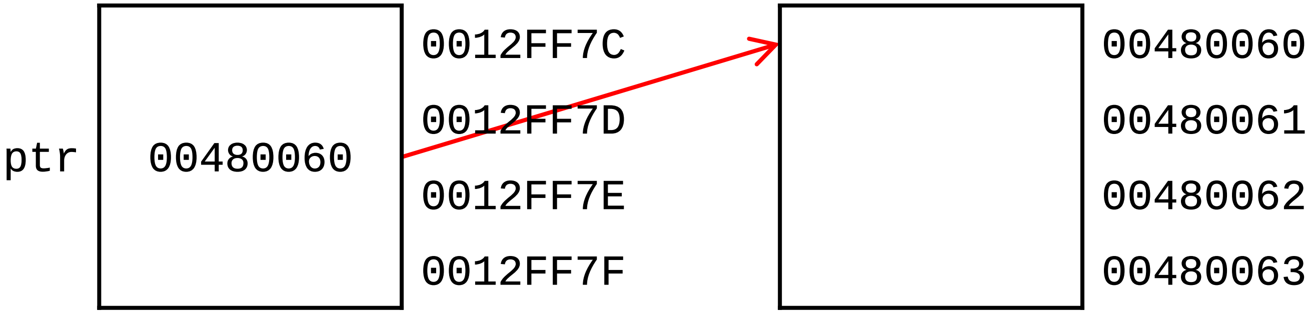
0012FF7D

0012FF7E

0012FF7F

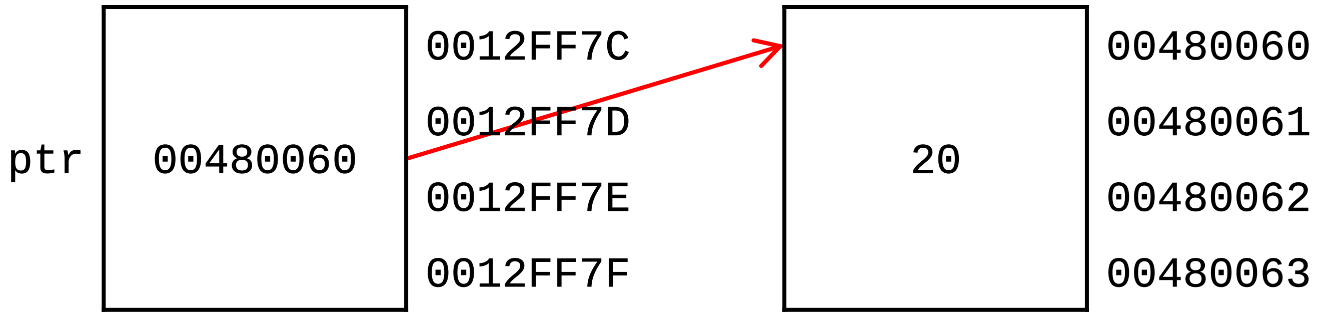
```
int main()
{
   // int *ptr = new int;
   int *ptr;
   ptr = new int;
   *ptr = 20;
   cout << *ptr << endl;
   cout << ptr << endl << endl;
   delete ptr;
}
```

Output



```
int main()
{
   // int *ptr = new int;
   int *ptr;
   ptr = new int;
   *ptr = 20;
   cout << *ptr << endl;
   cout << ptr << endl << endl;
   delete ptr;
}
```

Output



ptr

00480060

0012FF7C

0012FF7D

0012FF7E

0012FF7F

20

00480060

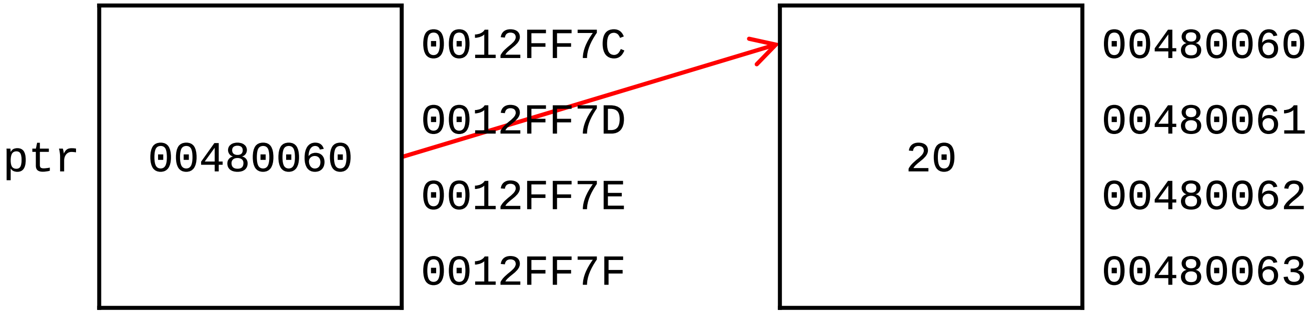
00480061

00480062

00480063

```
int main()
{
   // int *ptr = new int;
   int *ptr;
   ptr = new int;
   *ptr = 20;
   cout << *ptr << endl;
   cout << ptr << endl << endl;
   delete ptr;
}
```

Output



ptr 00480060

0012FF7C
0012FF7D
0012FF7E
0012FF7F

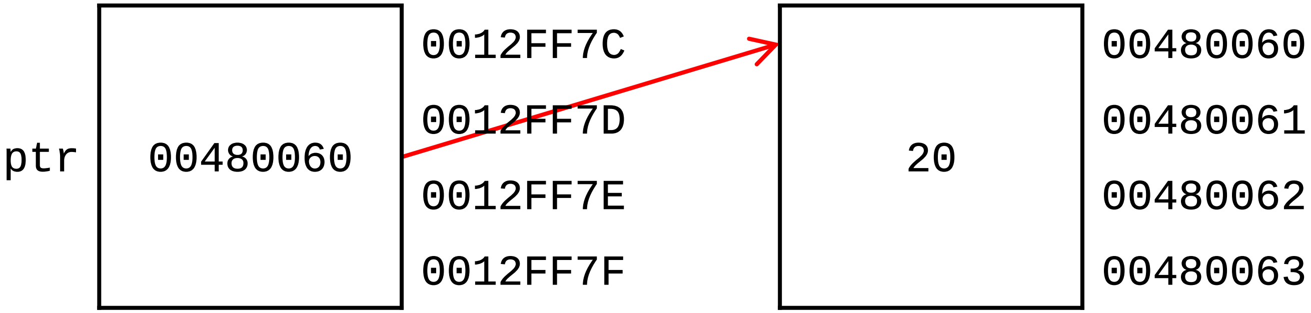
20

00480060
00480061
00480062
00480063

```
int main()
{
   // int *ptr = new int;
   int *ptr;
   ptr = new int;
   *ptr = 20;
   cout << *ptr << endl;
   cout << ptr << endl << endl;
   delete ptr;
}
```

Output

```
20
```

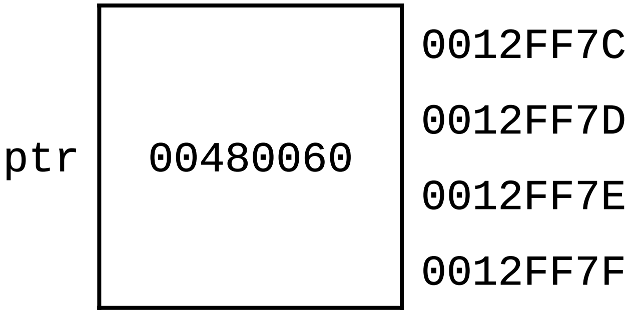


```
int main()
{
   // int *ptr = new int;
   int *ptr;
   ptr = new int;
   *ptr = 20;
   cout << *ptr << endl;
   cout << ptr << endl << endl;
   delete ptr;
}
```

Output

```
20

00480060
```



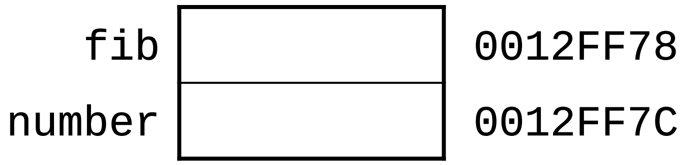
```
int main()
{
   // int *ptr = new int;
   int *ptr;
   ptr = new int;
   *ptr = 20;
   cout << *ptr << endl;
   cout << ptr << endl << endl;
   delete ptr;
}
```

Output

```
20

00480060
```

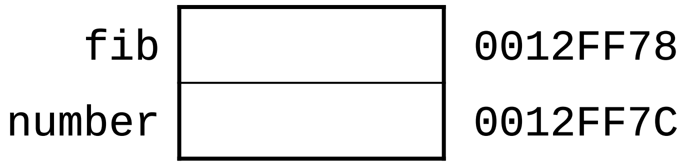
Dynamically allocate an array - Fibonacci numbers



fib 0012FF78

number 0012FF7C

```
int main()
{
   int number;
   cin >> number;
   int *fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



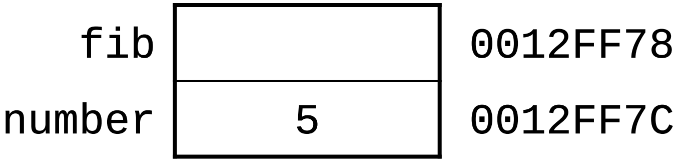
fib 0012FF78

number 0012FF7C

```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```

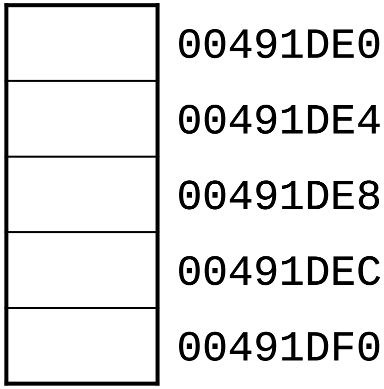
fib		0012FF78
number	5	0012FF7C

```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



fib 0012FF78

number 5 0012FF7C



00491DE0

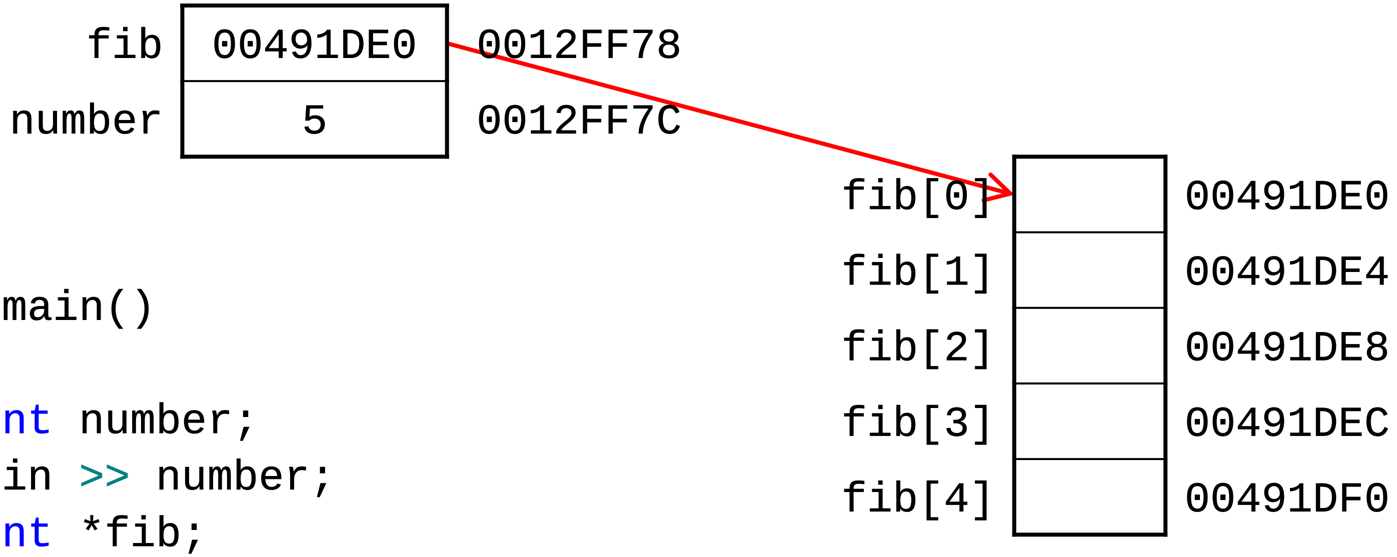
00491DE4

00491DE8

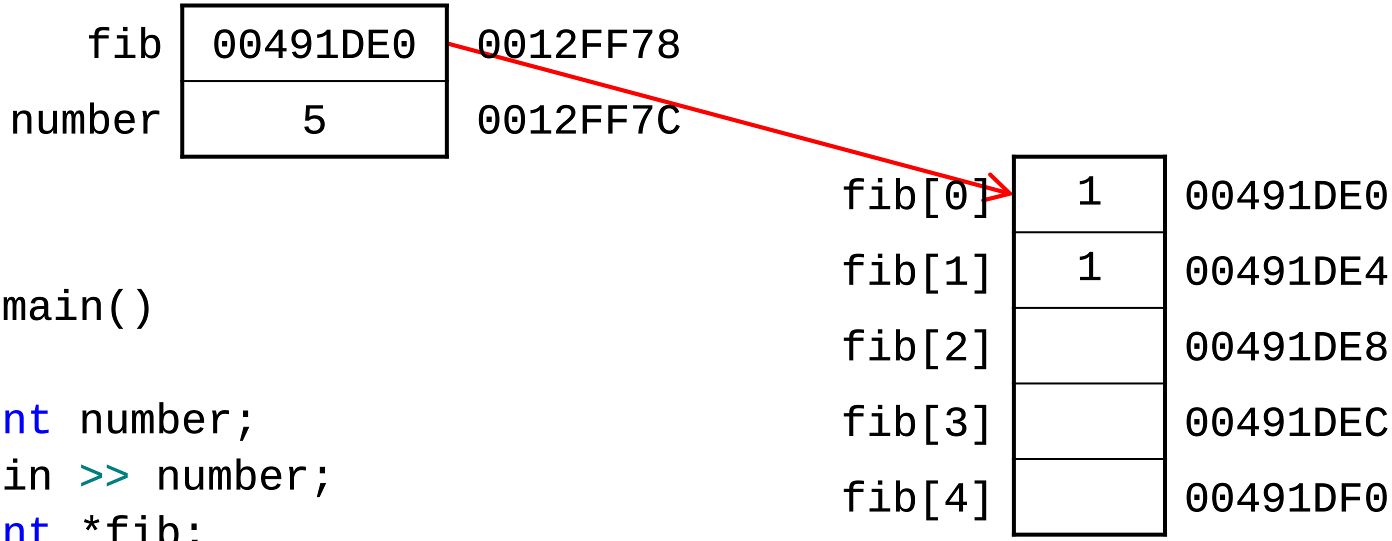
00491DEC

00491DF0

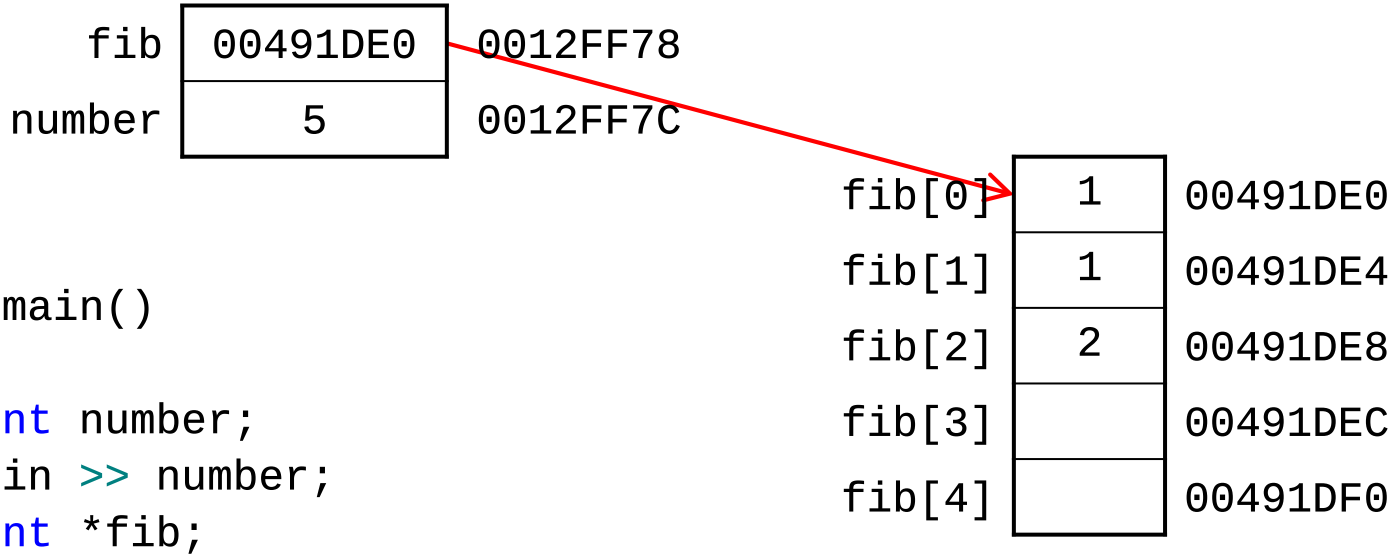
```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



fib 00491DE0 0012FF78

number 5 0012FF7C

fib[0] 1 00491DE0

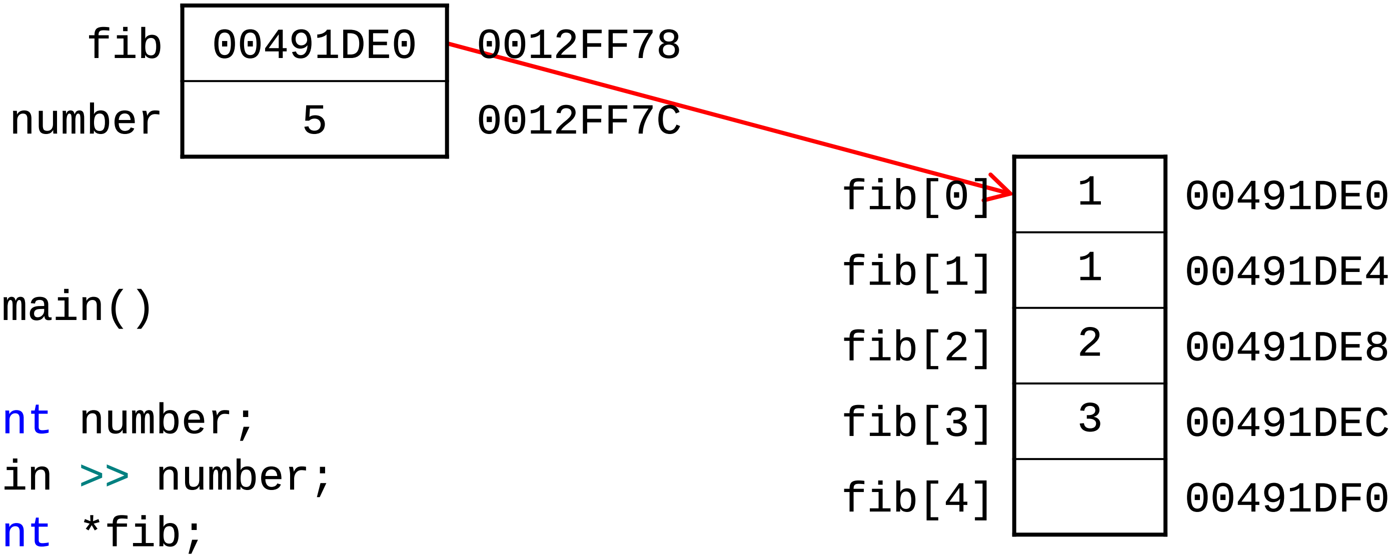
fib[1] 1 00491DE4

fib[2] 2 00491DE8

fib[3] 00491DEC

fib[4] 00491DF0

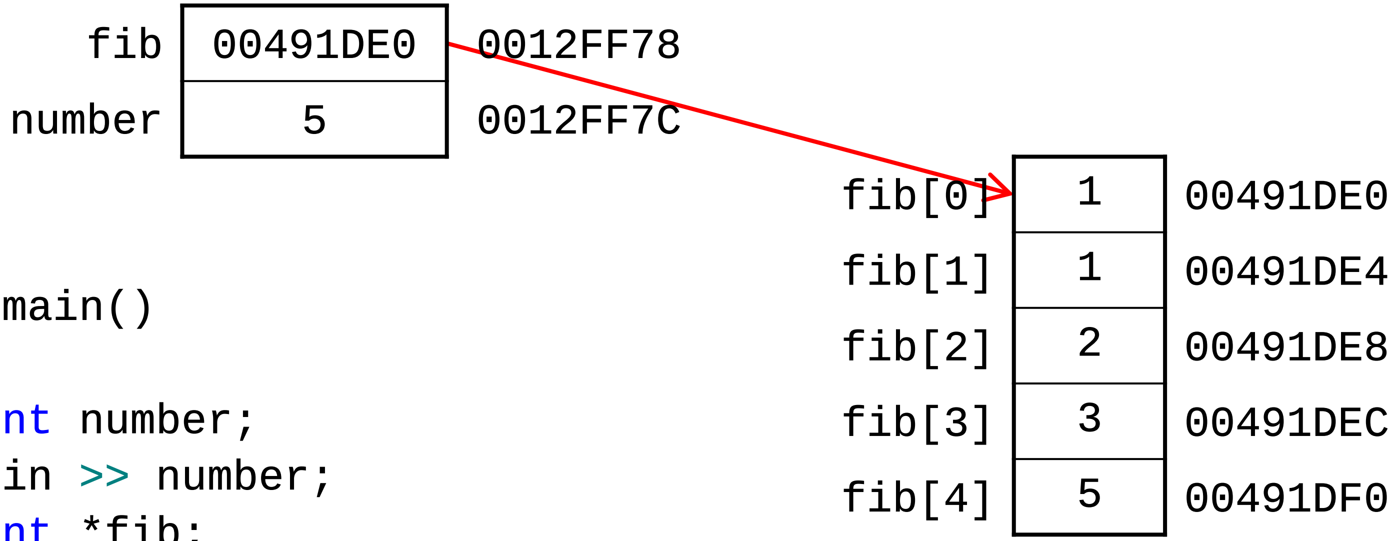
```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



fib	00491DE0	0012FF78
number	5	0012FF7C

fib[0]	1	00491DE0
fib[1]	1	00491DE4
fib[2]	2	00491DE8
fib[3]	3	00491DEC
fib[4]		00491DF0

```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



fib 00491DE0 0012FF78

number 5 0012FF7C

fib[0] 1 00491DE0

fib[1] 1 00491DE4

fib[2] 2 00491DE8

fib[3] 3 00491DEC

fib[4] 5 00491DF0

```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```

fib	00491DE0	0012FF78
number	5	0012FF7C

```
int main()
{
   int number;
   cin >> number;
   int *fib;
   fib = new int[ number ];
   fib[ 0 ] = 1;
   fib[ 1 ] = 1;
   for( int i = 2; i < number; i++ )
      fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
   for( int i = 0; i < number; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```

Passing dynamically allocated arrays to functions

```
int main()
{
   int n;
   cout << "Enter a positive number ( 2 - 46 ): ";
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[i] << endl;
   delete [] fib;
}

void fibonacci( int n, int *f )
{
   f[0] = 1;
   f[1] = 1;
   for( int i = 2; i < n; i++ )
      f[i] = f[i-2] + f[i-1];
}
```

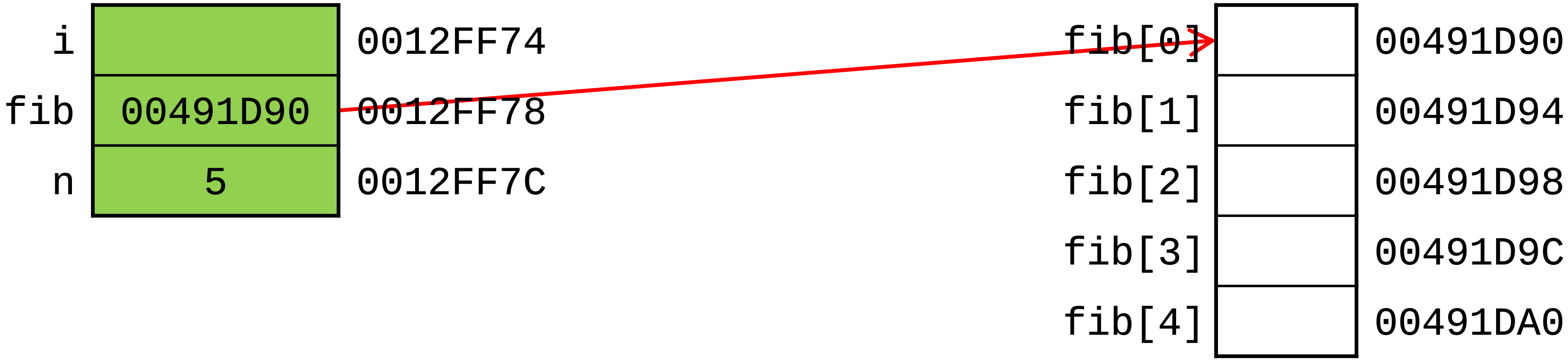
```
int main()
{
   int n;
   cout << "Enter a positive number ( 2 - 46 ): ";
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}

void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

i		0012FF74
fib		0012FF78
n	5	0012FF7C

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```

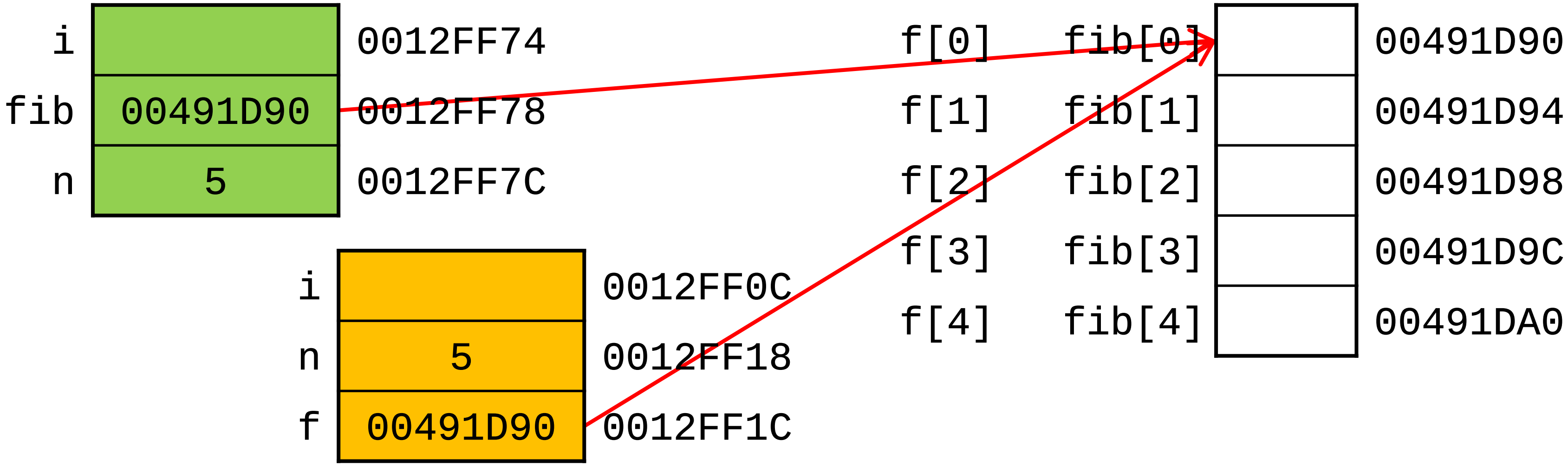


i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

fib[0]		00491D90
fib[1]		00491D94
fib[2]		00491D98
fib[3]		00491D9C
fib[4]		00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



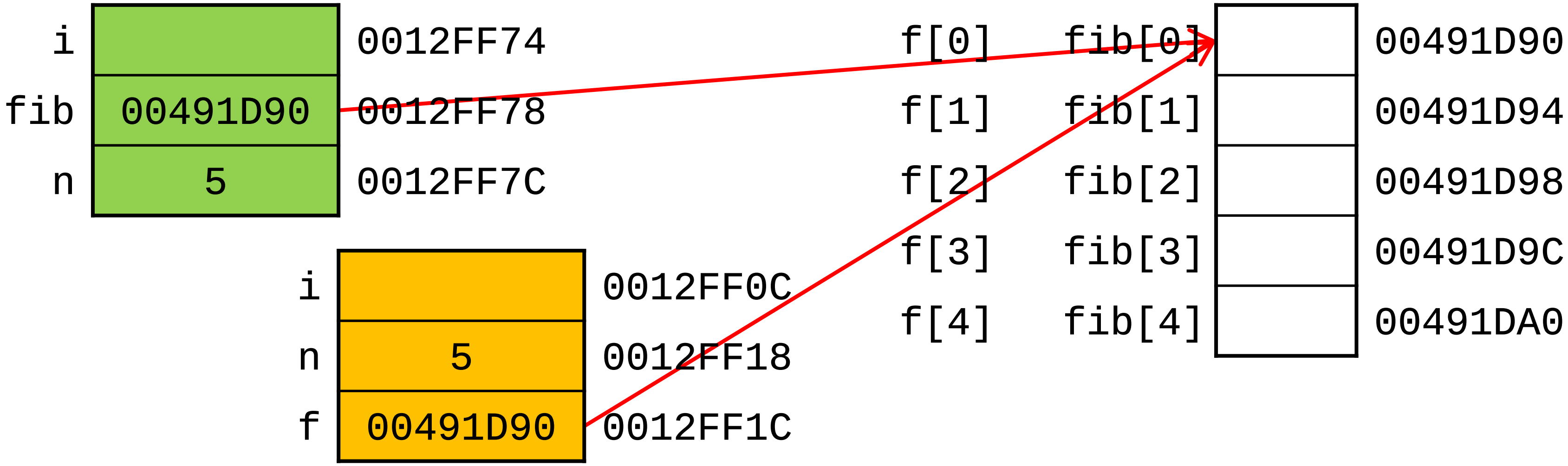
i 0012FF74
fib 00491D90 0012FF78
n 5 0012FF7C

i 0012FF0C
n 5 0012FF18
f 00491D90 0012FF1C

f[0] fib[0] 00491D90
f[1] fib[1] 00491D94
f[2] fib[2] 00491D98
f[3] fib[3] 00491D9C
f[4] fib[4] 00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



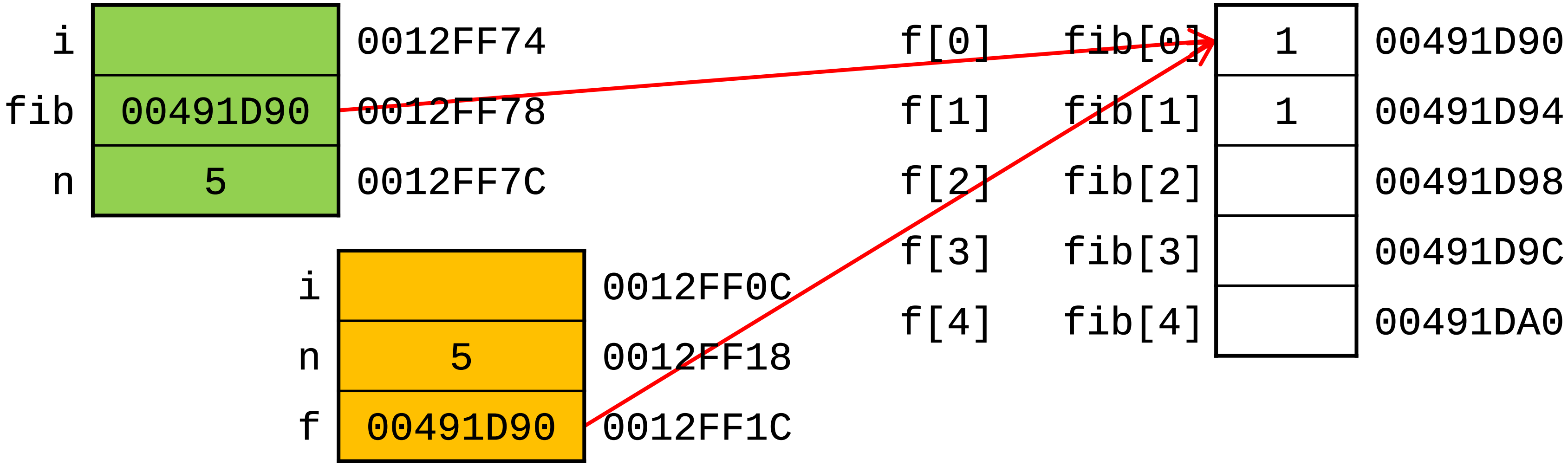
i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

i		0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

f[0]	fib[0]		00491D90
f[1]	fib[1]		00491D94
f[2]	fib[2]		00491D98
f[3]	fib[3]		00491D9C
f[4]	fib[4]		00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



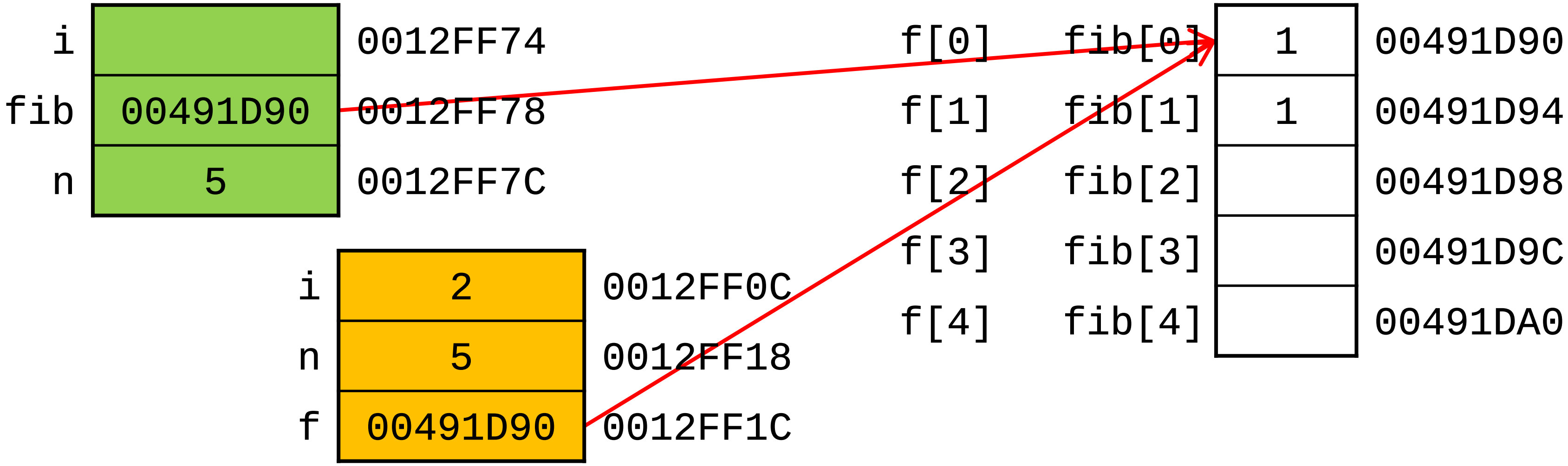
i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

i		0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

f[0]	fib[0]	1	00491D90
f[1]	fib[1]	1	00491D94
f[2]	fib[2]		00491D98
f[3]	fib[3]		00491D9C
f[4]	fib[4]		00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



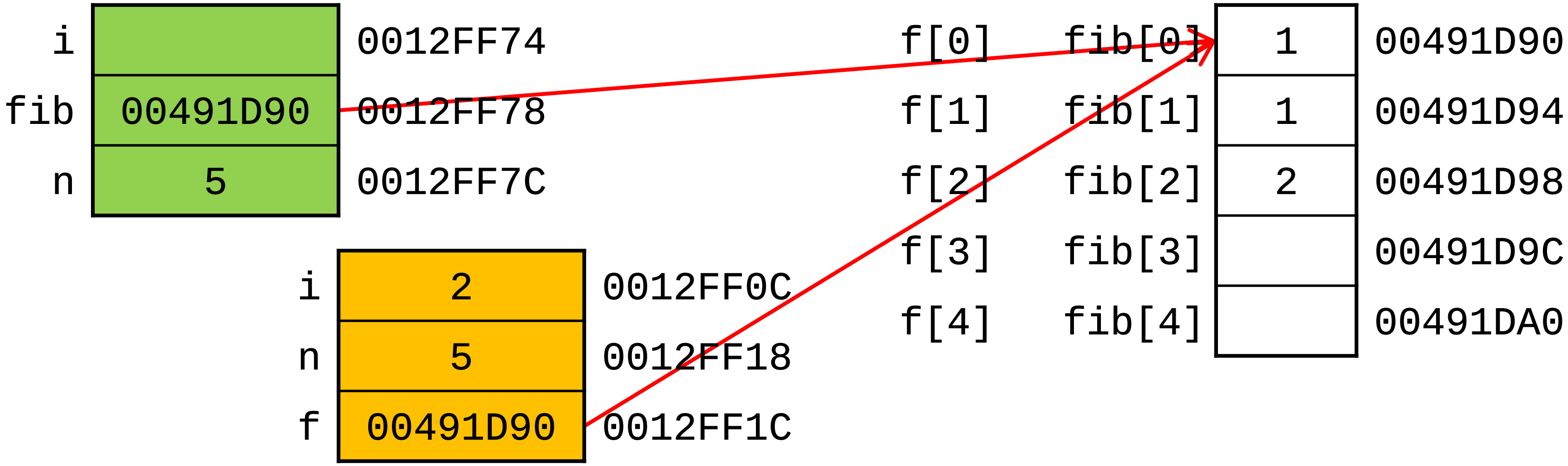
i | | 0012FF74
fib | 00491D90 | 0012FF78
n | 5 | 0012FF7C

i | 2 | 0012FF0C
n | 5 | 0012FF18
f | 00491D90 | 0012FF1C

f[0] fib[0] 1 00491D90
f[1] fib[1] 1 00491D94
f[2] fib[2] 00491D98
f[3] fib[3] 00491D9C
f[4] fib[4] 00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



i | | 0012FF74

fib | 00491D90 | 0012FF78

n | 5 | 0012FF7C

i | 2 | 0012FF0C

n | 5 | 0012FF18

f | 00491D90 | 0012FF1C

f[0] fib[0] 1 00491D90

f[1] fib[1] 1 00491D94

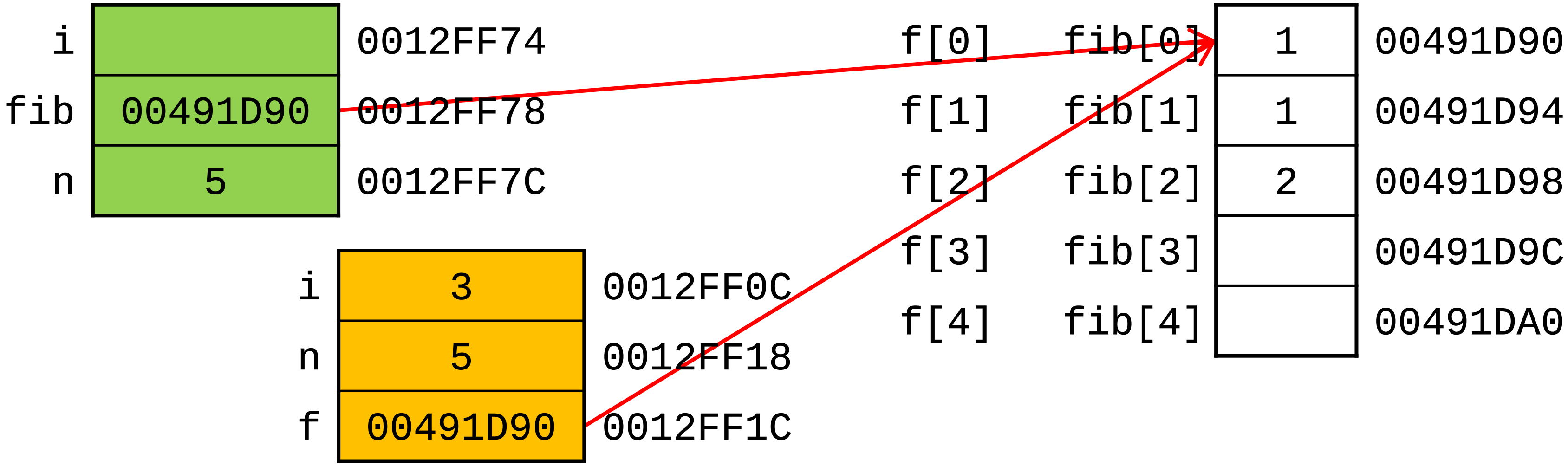
f[2] fib[2] 2 00491D98

f[3] fib[3] 00491D9C

f[4] fib[4] 00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



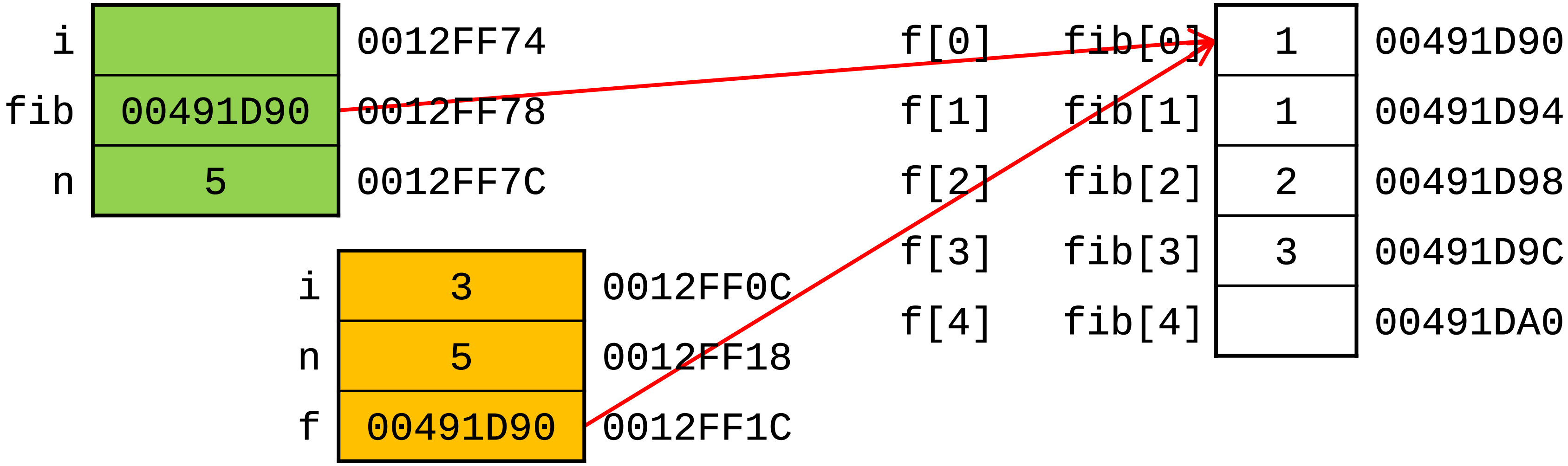
i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

i	3	0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

f[0]	fib[0]	1	00491D90
f[1]	fib[1]	1	00491D94
f[2]	fib[2]	2	00491D98
f[3]	fib[3]		00491D9C
f[4]	fib[4]		00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



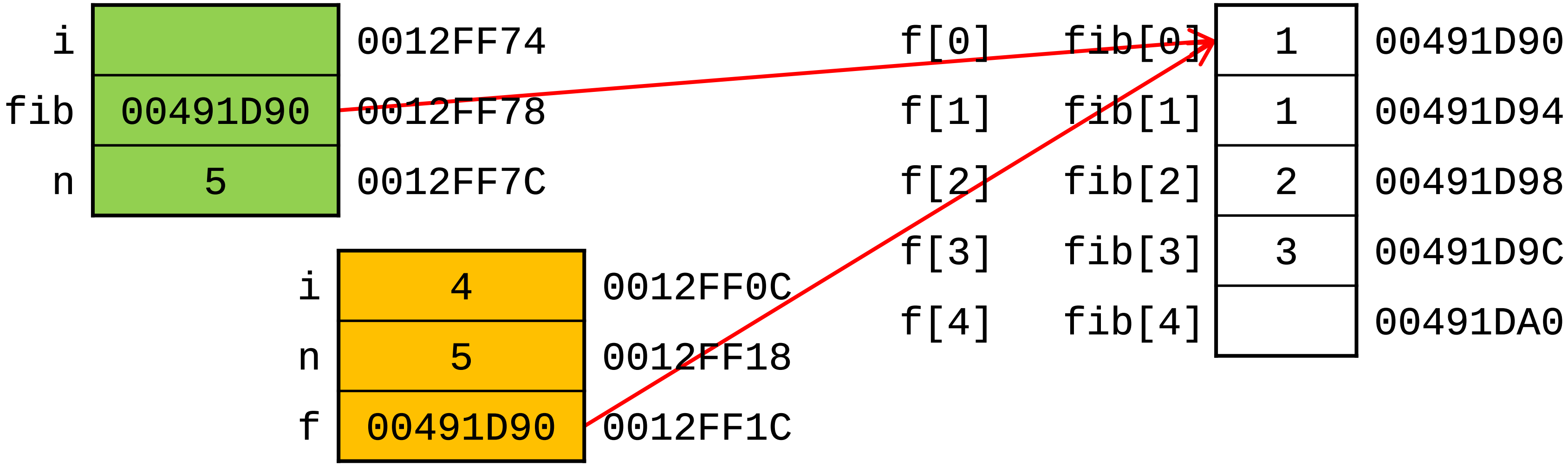
i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

i	3	0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

f[0]	fib[0]	1	00491D90
f[1]	fib[1]	1	00491D94
f[2]	fib[2]	2	00491D98
f[3]	fib[3]	3	00491D9C
f[4]	fib[4]		00491DA0

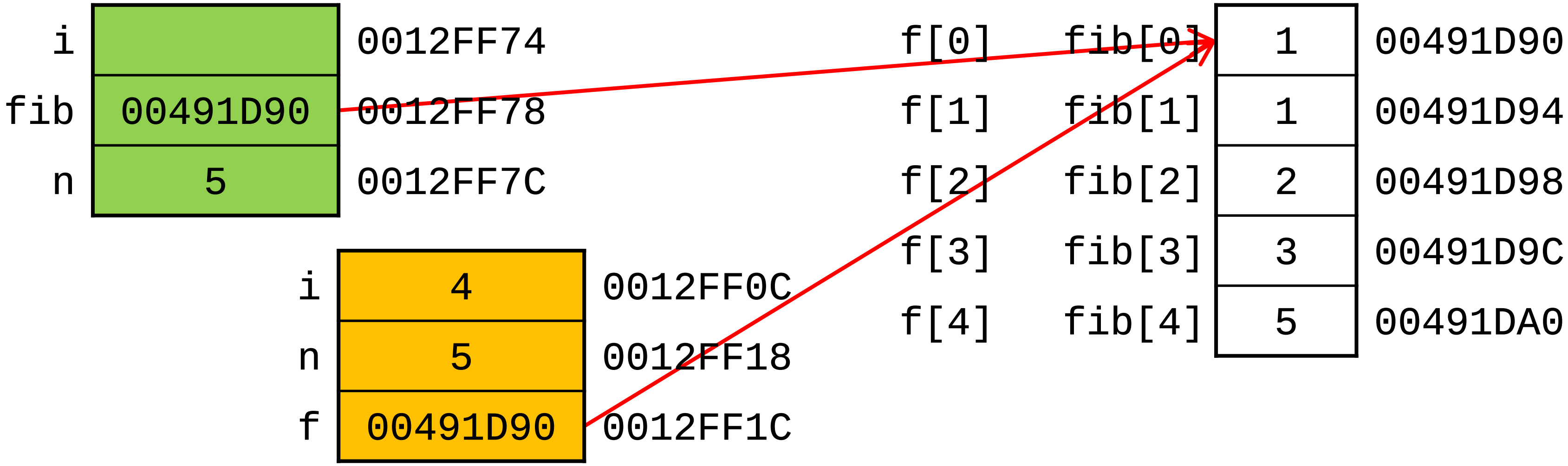
```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



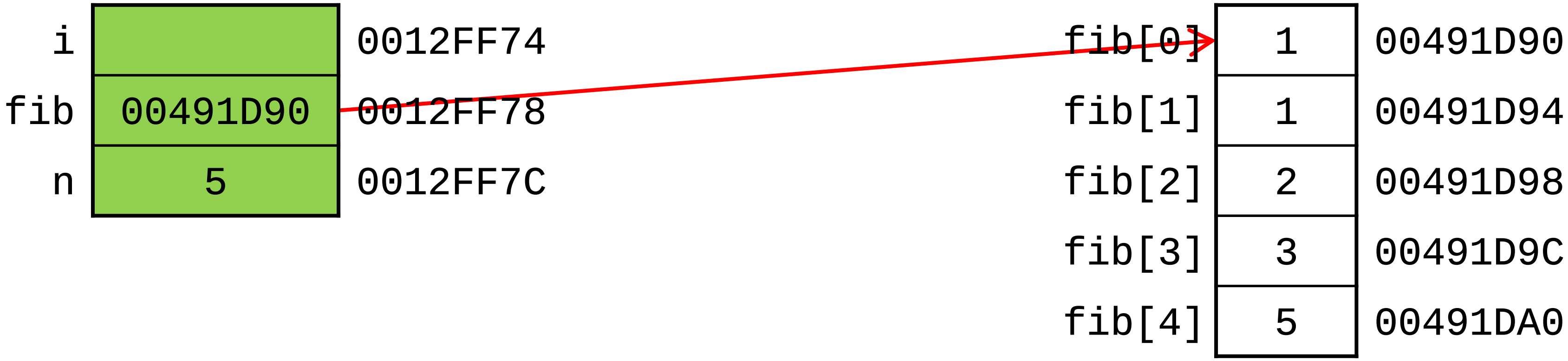
i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

i	4	0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

f[0]	fib[0]	1	00491D90
f[1]	fib[1]	1	00491D94
f[2]	fib[2]	2	00491D98
f[3]	fib[3]	3	00491D9C
f[4]	fib[4]	5	00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```



i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

fib[0]	1	00491D90
fib[1]	1	00491D94
fib[2]	2	00491D98
fib[3]	3	00491D9C
fib[4]	5	00491DA0

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```

i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

```
void fibonacci( int n, int *f )
{
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
   int n;
   cin >> n;
   int *fib = new int[ n ];
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}
```

Returns dynamically allocated arrays back to the calling function

```
int main()
{
   int n = 0;
   int *fib = nullptr;
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[i] << endl;
   delete [] fib;
}

void fibonacci( int n, int *f )
{
   cout << "Enter a positive number ( 2 - 46 ): ";
   cin >> n;
   f = new int[ n ]
   f[0] = 1;
   f[1] = 1;
   for( int i = 2; i < n; i++ )
      f[i] = f[i-2] + f[i-1];
}
```

```
int main()
{
   int n = 0;
   int *fib = nullptr;
   fibonacci( n, fib );
   for( int i = 0; i < n; i++ )
      cout << fib[ i ] << endl;
   delete[] fib;
}

void fibonacci( int n, int *f )
{
   cout << "Enter a positive number ( 2 - 46 ): ";
   cin >> n;
   f = new int[ n ];
   f[ 0 ] = 1;
   f[ 1 ] = 1;
   for( int i = 2; i < n; i++ )
      f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```